

**RAPPORT**  
**Chef d'Oeuvre**  
**Conception**

COURDY-BAHSOUN CLÉMENCE - DEKER SYLVAIN  
KOTTATH SANDEEP - MOUSSA NAHOR

Groupe : Ré-id

M2 IGAI

## 1 Introduction

Le projet proposé par le client, M. Alain Crouzil à pour but le développement d'une interface permettant de faire de la ré-identification de véhicule dans un corpus de vidéo. Il s'agit de détecter les véhicules dans les vidéos et de les comparer au véhicule recherché afin de fournir une liste pertinente des vidéos pouvant contenir le véhicule recherché.

Pour la ré-identification de véhicule la méthode Fusion of Attributes and colors features (FACT) produit de bons résultats en temps et en précision. Elle repose sur le calcul de trois descripteurs de l'image (texture, couleur et sémantique)

La méthode BOW-SIFT permet de calculer les descripteurs SIFT des points d'intérêts d'une image, puis de les classifier sous la forme d'un dictionnaire par apprentissage non supervisé hiérarchique avec la méthode des k-moyenne, avec un nombre de classe fixé à 10000. L'histogramme du nombre de descripteurs par classe devient alors le descripteur de l'image.

La méthode BOW-CN permet de calculer un descripteur de couleur sous la forme d'un vecteur obtenu par le calcul d'un histogramme sur 250 classes obtenues par un algorithme de k-moyenne. Les descripteurs permettant de calculer l'histogramme sont les valeurs des canaux des pixels.

Le réseau de neurones convolutifs GoogleNet prend une image en entrée et en lui appliquant des convolutions il permet d'obtenir différents descripteurs qui permettent une caractérisation de l'image propre au réseau : les poids de la couche finale du réseau vont être utilisé comme descripteur sémantique du véhicule.

Ce rapport a pour but d'approfondir les bases posées dans le rapport de spécification en déterminant une vue détaillée du système afin de faciliter le travail de conception. Le planning et les risques sont également mis à jour pour une meilleure appréhension de la réalisation du projet.

## 2 Vue d'ensemble détaillée

Dans cette section nous allons définir plus en détails le système. La liste qui suit va donner plus de détail sur les blocs définis en indiquant : la signature des fonctions, les entrées et sorties. Les getters et setters ne sont pas détaillés, ils seront ajoutés au besoin.

1. Bloc FACT

- (a) *sift*

- i. Entrée : ImageVéhicule
  - ii. Sortie : descripteursSiftParPx : double[][][]
  - iii. fonction : calculer descripteur Sift de chaque pixel de l'image en entrée
- (b) *cn*
- i. Entrée : ImageVéhicule
  - ii. Sortie : descripteursCNParPx : double[][][]
  - iii. fonction : calculer descripteur CN de chaque pixel de l'image en entrée
- (c) *calculDescripteurSift*
- i. Entrée : ImageVéhicule
  - ii. Sortie : descripteurSift : double[][]
  - iii. fonction : calcul le descripteur Sift de l'image en entrée en appliquant la fonction bagOfWord(k=10000) sur les descripteurs Sift de chaque pixel
- (d) *calculDescripteurCN*
- i. Entrée : ImageVéhicule
  - ii. Sortie : descripteurCN : double[][]
  - iii. fonction : calculer le descripteur CN de l'image en entrée en appliquant la fonction bagOfWord(k=250) sur les descripteurs de couleur de chaque pixel
- (e) *calculDescripteurSem*
- i. Entrée : ImageVéhicule
  - ii. Sortie : descripteurSem : double[]
  - iii. fonction : calcul le descripteur sémantique de l'image en entrée en récupérant le vecteur de poids en sortie de l'avant dernière couche du réseau de neurone convolutif GoogleNet.
- (f) *calculDescripteurFact*
- i. Entrées : descripteurSift, descripteurCN, descripteurSem, coeffDescripteurSift, coeffDescripteurCN, coeffDescripteurSem
  - ii. Sortie : descripteurFact : double[].
  - iii. fonction : Passer les descripteurs dans le même repère et en faire la somme pondérée selon les coefficients en entrée.

## 2. Bloc Apprentissage

- (a) *BagOfWord*
    - i. Entrées : k le nombre de classe, liste des vecteurs de même taille à classer.
    - ii. Sortie : Histogramme des vecteurs : double [][]
    - iii. fonction : Classer les vecteurs par k-moyenne.
  - (b) Spécialisation de Yolo3
  - (c) Entraînement de GoogleNet
3. Bloc Détection
- (a) *detecteurYolo*
    - i. Entrées : corpus Video de surveillance
    - ii. Sortie : liste ImageVehicule
    - iii. fonction : détecter les véhicules avec le réseau de neurone Yolo3 entraîné à cet effet, et renseigner les propriétés de l'image.
  - (b) *construireCorpusFact*
    - i. Entrées : chemin du fichier, liste de ImageVehicule
    - ii. Sortie : 0 si fichier bien rédigé, erreur sinon.
    - iii. fonction : Calculer le vecteur du descripteurFact. Rédiger dans un fichier les propriétés de chaque image du corpus.
4. Bloc Ré-identification
- (a) *distanceEuclidienne*
    - i. Entrées : descripteur Fact de l'image recherchée, descripteur Fact d'une image du corpus
    - ii. Sortie : distance : double
    - iii. fonction : calculer le score par distance euclidienne au carrée des deux vecteurs en entrée.
  - (b) *distanceDescripteurs*
    - i. Entrées : corpusFact, descripteur Fact de l'image recherchée
    - ii. Sortie : distanceDescripteurs : list<idImageCorpus,score>
    - iii. fonction : calculer le score de distance du descripteur de l'image recherchée avec tous les descripteurs du corpus.
  - (c) *trierListeScore*
    - i. Entrées : liste<idImageCorpus,score>
    - ii. Sortie : liste<idImage,score> triée
    - iii. fonction : trier les scores par ordre croissant.

## 5. bloc IHM

### (a) *chargerImage*

- i. Entrées : cheminImage
- ii. Sortie : 0 si existe, erreur sinon
- iii. fonction : Charger image du véhicule, lui donner la bonne dimension.

### (b) *afficherResultat*

- i. Entrées : liste<id,score>, n
- ii. Sortie : n véhicules les plus ressemblant.
- iii. fonction : afficher les n images, leur video d'appartenance et l'instant de la video, ainsi que le score de ressemblance.

## 3 Structures et diagramme de classe

### 3.1 Structure ImageVehicule

Chaque image de véhicule détectée par Yolo3 aura sa structure de la forme suivante :

```
struct (  
int idIm, double[][] imageVehicule,  
string cheminVideo,  
double[] descripteurFact,  
Time instantApparition  
) ImageVehicule;
```

### 3.2

Le diagramme de classe qui suit est une vue de l'esprit. Le but était d'avoir un schéma plus détaillé que la répartition en module, permettant d'illustrer les liens, ainsi que les paramètres et les entrées sorties de chaque fonction.

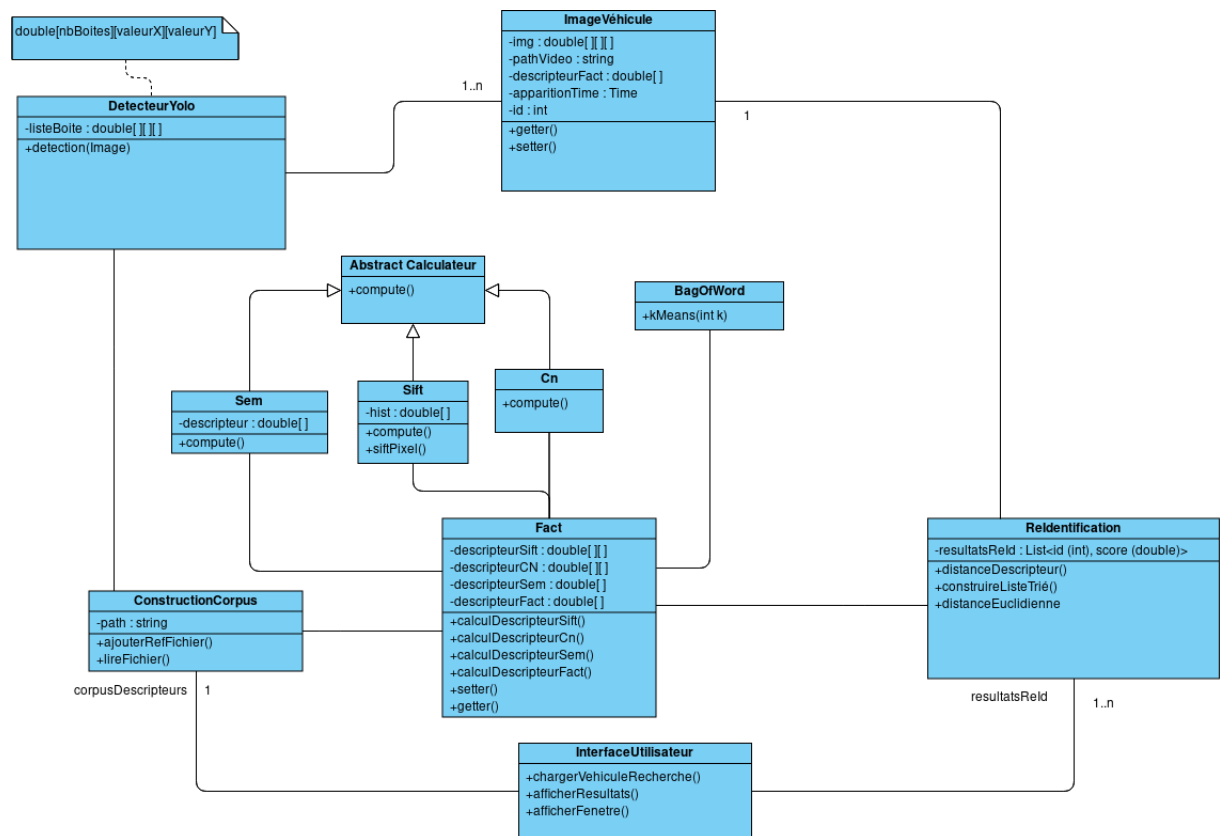


FIGURE 1 – Diagramme de classe

## 4 Test unitaires

Pour chaque module les tests seront effectués de manière indépendantes. La ré-identification de véhicules n’étant pas un sujet majeur de recherche, des jeux de données comportant des vérités de terrain n’existent pas. Pour cette raison nous ne pourrons pas automatiser les vérifications, il nous faudra directement voir si les résultats sont cohérents en fonction des données d’entrées.

### Bloc Détection

Il faut s’assurer par le biais d’un affichage que les boites englobent bien des voitures ou des camions.

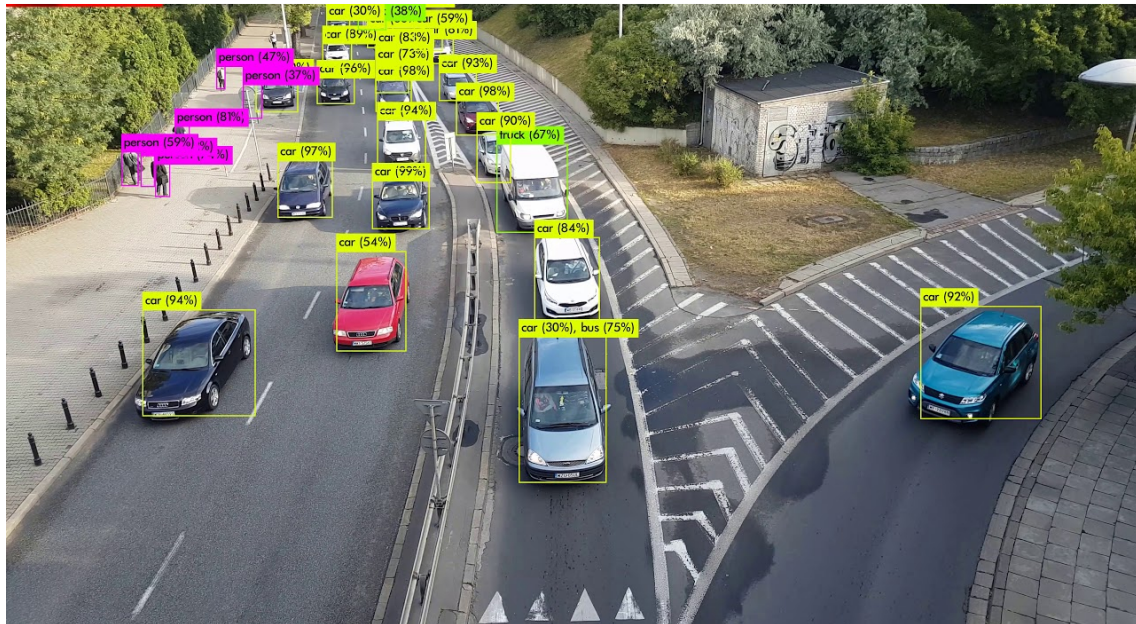


FIGURE 2 – Détection des véhicules

Les piétons et autres objets non pertinents sont à retirer de notre détection.

### Bloc Fact

Les méthodes SIFT et CN sont basées sur l'algorithme des K-Means qui ont leurs centroïdes initialisés de manière aléatoire. A cause de cela il n'est pas possible de savoir si une itération est valide cependant notre implémentation pourra être comparé avec d'autres algorithmes lors de test sur une même image en désactivant l'initialisation aléatoire des centroïdes. Pour le réseau de neurones GoogleNet nous ne pourrons que nous baser sur la classe renvoyée par le réseau ainsi que son taux de confiance, les poids des neurones seront inutilisables pour des vérifications.

### Bloc Ré-identification

Le calcul de distance entre le descripteur Fact et le corpus pourra être testé en créant un corpus contenant de fausses valeurs pour pallier à l'absence des données dans un premier temps.

### Bloc IHM

Le bloc IHM ne sera pas testé algorithmiquement mais pourra être évalué arbitrairement par le client.

## 5 Planning et Analyse de risque

### 5.1 Planning revu

#### Planning initial

	Janvier				Fevrier			
	1	2	3	4	5	6	7	8
1. Détection de véhicules	SK	SK						
2. Découpage des video en images			SK					
3. Récupération des boites englobant associés au images				SK				
4. Association/Indexation Véhicules-Video (Fichier)					SD			
5. K-Moyen hiérarchique des descripteurs					C	C		
6. Entraînement de GoogleNET	N	N	N					
7.a Calcul des descripteurs Sift (Texture)	C	C						
7.b Calcul des descripteurs CN (Couleur)	SD	SD						
7.c Réalisation d'un Bag-Of-Word pour Sift et CN			C+SD					
7.d Calcul de deux descripteurs (texture et couleurs) à partir de 7.a et 7.b avec les Bow en 7.c				C+SD				
7.e Récupération du descripteur de GoogleNET				N				
7.f Calcul du score FACT à partir des 3 descripteurs 7.d et 7.e					C			
8. Calcul du score FACT sur l'ensemble des véhicules contenues dans les boites englobante en 3.						SD		
9. Calcul de score FACT sur l'image du véhicule recherché						SD		
10. Associer le score FACT aux véhicules indexés dans le fichier en 4.						N		
11. Comparer le score de l'image du véhicule de référence aux autres ( distance SSD)							SD	
12. Trier les véhicules par ordre croissant							SD	
13.a Charger l'image du véhicule recherché							SD	
13.b Affichage des véhicules réidentifiés avec le nom de leur video et la score de distance calculée							N	

FIGURE 3 – Planning initial

Le planning initial prévoyait une répartition du travail sur les mois de janvier et de février. Ce planning n’a pas pu être suivi comme prévu, en effet, les objectifs prévus pour le mois de janvier n’ont pas pu être tenus dans les temps. La programmation des tâches a été construite avec une mauvaise connaissance du travail à fournir dans les autres matières. La répartition du travail a été considérée comme constante au cours de ces deux mois, alors que seul, le mois de janvier était chargé de cours avec une semaine de partiel. Les éléments de travail effectués jusqu’ici sont :

- La mise en place d’outil de collaboration : Git, Github.
- La mise en place de la structure de donnée principale de gestion d’image.
- L’étude d’une architecture globale du système dont la version la plus abouti est présenté dans ce document.

Ces éléments n’ont pas été considérés comme des tâches, mais ont été nécessaires pour la mise en place du projet. (voir figure 5.1).

#### Planning prévisionnel

Grâce à l’emploi du temps du cursus universitaire, il est possible d’affiner au mieux le nouveau planning prévisionnel. Autant le mois de janvier était pris par d’autres activités, autant le mois février est dédié à la réalisation du projet. C’est pourquoi la majeure partie de développement se porte désormais sur le seul mois de



février. D’ici la fin du mois de janvier, des partiels sont prévus et aucun travail de développement n’est sérieusement envisageable. Le nouveau planning prévisionnel apporte une modification majeure ; les itérations agiles passent de 1 semaines à 3 jours, chose qui tout à fait envisageable étant donné la possibilité de se consacrer à 100% sur le projet. Au moins Une réunion entre les membres de l’équipe sera faite à chaque fin d’itération. (voir figure 5.1).

	Fevrier										
	2	5	8	11	14	17	20	23	26		
1. Détection de véhicules	SK	SK									
2. Découpage des video en images			SK				SK				
3. Récupération des boites englobant associés au images				SK							
4. Association/Indexation Véhicules-Video (Fichier)					SD						
5. K-Moyen hierarchique des descripteurs						C		C			
6. Entraînement de GoogleNET	N	N	N								
7.a Calcul des descripteurs Sift (Texture)	C	C									
7. b Calcul des descripteurs CN (Couleur)	SD	SD									
7. c Réalisation d’un Bag-Of-Word pour Sift et CN			C+SD								
7. d Calcul de deux descripteurs (texture et couleurs) à partir de 7. a et 7. b avec les Bow en 7. c				C+SD							
7. e Récupération du descripteur de GoogleNET				N							
7. f Calcul du score FACT à partir des 3 descripteurs 7. d et 7. e					C						
8. Calcul du score FACT sur l’ensemble des véhicules contenues dans les boites englobante en 3.						SD					
9. Calcul de score FACT sur l’image du véhicule recherché							SD				
10. Associer le score FACT aux véhicules indexés dans le fichier en 4.							N				
11. Comparer le score de l’image du véhicule de référence aux autres ( distance SSD)								SD			
12. Trier les véhicules par ordre croissant									SD		
13. a Charger l’image du véhicule recherché								SD			
13. b Affichage des véhicules réidentifiés avec le nom de leur video et la score de distance calculée									N		

FIGURE 4 – Planning prévisionnel

## 5.2 Risques revus

### Risques :

Un certain nombres de risques ont été répertoriés et classés à l’aide de la matrice de risque 5.2.

1. Facteur humain : mésentente et problème de communication au sein de l’équipe. Disparition d’un membre. Baisse de motivation.
2. Mauvaise évaluation du temps alloué aux tâches.
3. Répartition des tâches.
4. Problèmes d’assemblage des différents modules entre eux.
5. Relation avec le client.
6. Retard sur le planning initial à cause des examens

### Solution :

1. Anticiper en partant sur des attentes minimales avec le client.

GRAVITE \ PROBABILITE	Mineure	Significative	Grave	Très grave
Fréquent		5	1	2
Peu fréquent		6	3	
Rare			4	
Très rare				

FIGURE 5 – Matrice des risques

2. Réunion d'équipe régulière pour voir l'avancement et mettre à jour le planning.
3. Réunion d'équipe régulière pour parler des difficultés et des réussites rencontrés afin d'avancer le plus efficacement possible.
4. Suivre les spécifications et les mêmes normes afin de faciliter l'intégration des modules.
5. S'assurer avec des rencontres régulières de la convergence entre le travail réalisé et les attentes du client.
6. Revoir la répartition des tâches pour la période consacrée au chef d'oeuvre.

# Bibliographie

- [1] Xinchun Liu, Wu Liu, Huadong Ma, Huiyuan Fu. *Large-scale vehicle re-identification in urban surveillance videos*. ICME, 1–6, 2016.
- [2] Joseph Redmon, Ali Farhadi. *YOLOv3 : An Incremental Improvement*. arXiv/CoRR, 2018.
- [3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. *SSD : Single Shot MultiBox Detector*. ECCV, 21–37, 2016.
- [4] Linjie Yang, Ping Luo, Chen Change Loy, Xiaoou Tang. *A Large-Scale Car Dataset for Fine-Grained Categorization and Verification*. arXiv/CoRR, 2015.
- [5] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. *Going Deeper with Convolutions*.  
<https://arxiv.org/abs/1409.4842>